

**APPLICATION FOR  
UNITED STATES PATENT  
IN THE NAME OF**

**DANIEL J. MENDEZ**

OF

SYSTEM AND METHOD FOR USING A WORKSPACE DATA MANAGER  
TO ACCESS, MANIPULATE AND SYNCHRONIZE NETWORK DATA

**DOCKET NO. 40827.00002**

**Please direct communications to:**

**Intellectual Property Department**  
**Graham & James LLP**  
**600 Hansen Way**  
**Palo Alto, CA 94304-1043**  
**(650) 856-6500**

**Express Mail Number** **EM320545362US**

090806Z JUL 74

SYSTEM AND METHOD FOR USING A WORKSPACE DATA MANAGER  
TO ACCESS, MANIPULATE AND SYNCHRONIZE NETWORK DATA

PRIORITY REFERENCES

- 5 This application claims priority of and hereby incorporates by reference
- U.S. patent <sup>pending</sup> application serial number 08/766,307, entitled "System and Method for
- Globally Accessing Computer Services," filed on December 13, 1996, by
- inventors Mark D. Riggins, et al; U.S. patent <sup>pending</sup> application serial number
- 08/841,950, entitled "System and Method for Enabling Secure Access to Services
- 10 in a Computer Network, filed on April 8, 1997, by inventor Mark D. Riggins; U.S. <sup>pending</sup>
- patent application serial number 08/865,075, <sup>and now U.S. Patent No 6,023,708 4-6-00 dm</sup> entitled "System and Method for
- Using a Global Translator to Synchronize Workspace Elements Across a
- Network," filed on May 29, 1997, by inventors Daniel J. Mendez, et al.; U.S. <sup>pending</sup>
- patent application serial number 08/835,997, entitled "System and Method for
- 15 Securely Synchronizing Multiple Copies of a Workspace Element in a Network,"
- filed on April 11, 1997, by inventors Daniel J. Mendez, et al.; U.S. <sup>pending</sup> patent
- application serial number 08/897,888, <sup>and now US Patent Number 5,961,590 4-6-00 dm</sup> entitled "System and Method for
- Synchronizing Electronic Mail Across a Network," filed on July 22, 1997, by
- ~~and now US Patent Number 5,970,590 4-6-00 dm~~
- inventors Daniel J. Mendez, et al.; U.S. patent application serial number
- 20 08/899,277, entitled "System and Method for Using an Authentication Applet to
- Identify and Authenticate a User in a Computer Network," filed on July 23, 1997,

by inventor Mark D. Riggins; and U.S. <sup>pending</sup> patent application serial number

08/903,118, entitled "System and Method for Globally and Securely Accessing

Unified Information in a Computer Network," filed on July 30, 1997, by inventors

Daniel J. Mendez, et al.

5

## BACKGROUND OF THE INVENTION

### 1. Field of the Invention

This invention relates generally to computer networks, and more particularly provides a system and method for using a workspace data manager to access network data.

### 2. Description of the Background Art

Data accessibility and consistency are significant concerns for computer users. For example, when a roaming user, i.e., a user who travels to a remote location, needs to review or manipulate data such as an e-mail or prepared document, the roaming user must either carry the data to the remote location or access a workstation remotely. Maintaining a true copy of a database is a cumbersome process. Accordingly, system designers have developed an array of techniques for connecting a remote terminal across a computer network to the workstation storing the data.

To guarantee readability of the downloaded data, the user must carry a laptop computer containing all the applications needed to present and enable manipulation of the downloaded data, or find a network-connected computer that contains the needed application programs. Further, when maintaining multiple  
5 independently modifiable copies of particular data, a user risks using an outdated version. By the time the user notices an inconsistency, interparty miscommunication or data loss may already have resulted. The user must then spend more time reconciling the inconsistent versions.

The problems of data accessibility and inconsistency are exacerbated when  
10 multiple copies of a document are maintained at different network locations. For example, due to network security systems such as conventional firewall technology, a user may have access only to a particular one of these network locations. Without access to the other sites, the user cannot confirm that the version on the accessible site is the most recent draft.

SUMMARY OF THE INVENTION

The present invention provides a system for using a workspace data manager to access, manipulate and synchronize workspace data. A workspace data manager may include a Personal Information Manager (PIM), a word processing program, a spreadsheet program, or any application program that enables manipulation of workspace data. Workspace data includes at least one workspace element, such as an e-mail, a day of calendar data, a word document, a bookmark, a sheet of spreadsheet data, or a portion thereof. Workspace data may include e-mails, calendar data, word documents, bookmarks, spreadsheet data, or portions thereof.

The system includes a communications module for downloading workspace data from a remote site, an application program interface coupled to the communications module for communicating with a workspace data manager to enable manipulation of the downloaded workspace data and thereby create manipulated data, and a general synchronization module coupled to the communications module for synchronizing the manipulated data with the workspace data stored at the remote site. An instantiator requests the workspace data manager to provide an interface for enabling manipulation of the downloaded workspace data. The workspace data manager may create another instance of the interface or may provide access to its only interface to enable manipulation of the data. A data reader translates the downloaded workspace data from the format

used by the remote site to the format used by the workspace data manager. For example, data stored at the global server site in a canonical format may be translated to Organizer™, Outlook™ or other workspace element manager format. Upon logout, a de-instantiator initiates synchronization and deletes the data stored locally. It will be appreciated that the system handles the situation where the data stored at the remote site has not changed and therefore includes the downloaded data, and the situation the data stored at the remote site has been modified and therefore is different than the downloaded data.

The present invention further provides a method of using a workspace data manager to enable access, manipulate and synchronize workspace data. The method comprises the steps of downloading data from a remote site, requesting a workspace data manager to enable manipulation of the data and thereby create manipulated data, and synchronizing the manipulated data with the data stored at the remote site.

The system and method of the present invention advantageously enable the use of an integral interface, instead of using an interface for the synchronization software, an interface for the workspace data manager and an interface for the communication engine downloading the workspace data. Accordingly, the user need not become familiar with multiple interfaces. The user need only find a remote site that includes a workspace data manager that includes assistant-like functionality. Assistant-like functionality includes services for interfacing

6

between the workspace data manager and the global server. Because the system and method substitute the global data for the local data or create an instance for the global data, the system and method further advantageously enable a workspace data manager to provide an interface for manipulating workspace data without

5    compromising the local data.

Further, the system and method advantageously provide a simple graphical user interface for enabling borrowing of the workspace data manager and synchronization of manipulated data. The system and method also advantageously delete downloaded data and all interfaces from the local client, so that no traces are

10    left on the local client for unprivileged users to review. Using the technology described in the applications incorporated by reference above, the system and method of the present invention further enable access and synchronization of data across different workspace data manager formats and across network firewalls.

7

09008354-011698

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating a network system, in accordance with the present invention;

FIG. 2 is a block diagram illustrating details of the home or work client of  
5 FIG. 1;

FIG. 3 is a block diagram illustrating details of the global server of FIG. 1;

FIG. 4 is a block diagram illustrating details of the remote client of FIG. 1;

FIG. 5 is a block diagram illustrating details of an assistant of FIG. 1;

FIG. 6 illustrates a personal information manager interface;

10 FIG. 7 illustrates a second personal information manager interface  
incorporating an assistant interface;

FIG. 8 is a flowchart illustrating a method of accessing network data from a remote site in accordance with the present invention; and

FIG. 9 is a flowchart illustrating a method of synchronizing network data  
15 from a remote site.





DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 is a block diagram illustrating a network system 100 for using a workspace data manager to access, manipulate and synchronize workspace data in accordance with the present invention. A workspace data manager may include a

5 Personal Information Manager (PIM), a word processing program, a spreadsheet program, or any application program that enables manipulation of workspace data. Workspace data includes at least one workspace element, such as an e-mail, a day of calendar data, a word document, a bookmark, a sheet of spreadsheet data, or a portion thereof. Workspace data may include e-mails, calendar data, word

10 documents, bookmarks, spreadsheet data, or portions thereof. Although the network system 100 is described with reference to PIM's, one skilled in the art will recognize that the system 100 will work with any workspace data manager.

Network system 100 includes a global server 105 coupled via a computer network 125 to a work client 110, to a home client 115 and to a remote client 120.

15 The global server 105 includes a synchronization agent 130 and workspace data 135. The work client 110 includes a base system 140 and workspace data 145. The home client 115 includes a base system 150 and workspace data 155.

Each of the base system 140 and the base system 150 cooperate with the synchronization agent 130 to synchronize workspace data 135, workspace data 145

20 and workspace data 155 between the work client 110, the home client 115 and the global server 105. Synchronization of workspace data 135, 145 and 155 is

described in detail in the patent applications incorporated by reference above.

However, a brief example of synchronization is provided for completeness.

First, the base system 140 on the work client 110 site negotiates a secure communications channel via any firewalls with the synchronization agent 130, for example, using Secure Sockets Layer (SSL) technology. The base systems 140 examines version information and if necessary the content of a workspace to determine the most updated version. The most updated version is then stored at the client 110 site and at the global server 105 site. The base system 140 repeats these operations for all workspace elements selected for synchronization. Second, the base system 150 on the home client 115 site uses similar steps to synchronize its workspace data 155 with the workspace data 135 on the global server 105 site. Accordingly, the most updated versions of the workspace data 135, 140 and 145 are stored at all three sites.

Each of the work client 110, the home client 115 and the remote client 120 includes a respective workspace data manager, e.g., a Personal Information Manager (PIM) 160, 165 and 170 such as Outlook™ 98 developed by Microsoft Corporation, Organizer 97 developed by Lotus Development Corporation or Sidekick 98 developed by Starfish Software. Each PIM 160, 165 and 170 includes an assistant 175, 180 and 185 that adds data access and synchronization functions to the PIM 160, 165 and 170. Accordingly, a user can transparently use an assistant 175, 180 or 185 via a PIM 160, 165 or 170 to access workspace data 135

10

from the global server 105, to present and enable manipulation of downloaded workspace data 135, and to synchronize manipulated downloaded data 135 with the workspace data 135 stored on the global server 105. Components and operations of the assistant 175, 180 or 185 are described in detail with reference to  
5 FIGs. 7-9.

FIG. 2 is a block diagram illustrating details of a data-synchronizing client 200, in a generic embodiment which exemplifies each of the work client 110 and the home client 115. The client 200 includes a processor 205, such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a  
10 communications channel 210. The client 200 further includes an input device 215 such as a keyboard and mouse, an output device 220 such as a Cathode Ray Tube (CRT) display, data storage 230 such as a magnetic disk, and internal storage 235 such as Random-Access Memory (RAM), each coupled to the communications channel 210. A communications interface 225 couples the communications  
15 channel 210 to the computer network 125.

An operating system 240 controls processing by processor 205, and is typically stored in data storage 230 and loaded into internal storage 235 (as illustrated) for execution. A base system 250, which cooperates with the synchronization agent 130 for synchronizing local workspace data 245 with  
20 workspace data 135, also may be stored in data storage 230 and loaded into internal storage 235 (as illustrated) for execution by processor 205. The local

N

workspace data 245 exemplifies workspace data 145 or workspace data 150, and may be stored in data storage 230.

5 A PIM 255 includes an assistant 260, which enables a user to download workspace data 135 from the global server 105, and to use the PIM 255 for displaying and manipulating the workspace data 135. The assistant 260 further enables the PIM 255 to synchronize the manipulated data 135 with the workspace data 135 on the global server 105. The PIM 255 exemplifies each of the PIM 160 on the work client 110 and the PIM 165 on the home client 115. The assistant 260 exemplifies each of the assistant 175 on the work client 110 and the assistant 180  
10 on the home client 115. The PIM 255 may be stored in data storage 230, and loaded into internal storage 235 (as illustrated) for execution by the processor 205.

One skilled in the art will recognize that the system 100 may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across  
15 a hardware channel, the Internet or an intranet, etc. One skilled in the art will also recognize that the programs and data may be received by and stored in the system 100 in alternative ways. For example, a computer-readable storage medium (CRSM) reader 265 such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus 210 for reading a  
20 computer-readable storage medium (CRSM) 270 such as a magnetic disk, a hard

disk, a magneto-optical disk, RAM, etc. Accordingly, the system 100 may receive programs and data via the CRSM reader 265.

FIG. 3 is a block diagram illustrating details of the global server 105. The global server 105 includes a processor 305, such as an Intel Pentium® microprocessor or a Motorola Power PC® microprocessor, coupled to a communications channel 310. The global server 105 further includes an input device 315 such as a keyboard and mouse, an output device 320 such as a CRT display, data storage 325 such as a magnetic disk, and internal storage 330 such as RAM, each coupled to the communications channel 310. A communications interface 325 couples the communications channel 310 to the computer network 125.

An operating system 340 controls processing by processor 305, and is typically stored in data storage 330 and loaded into internal storage 335 (as illustrated) for execution. The synchronization agent 130, which cooperates with the base system 250 (FIG. 2) for synchronizing local workspace data 245 with workspace data 135, also may be stored in data storage 330 and loaded into internal storage 335 (as illustrated) for execution by processor 305. The workspace data 135 may be stored in data storage 230.

One skilled in the art will recognize that the system 100 may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across

a hardware channel, the Internet or an intranet, etc. One skilled in the art will also recognize that the programs and data may be received by and stored in the system 100 in alternative ways. For example, a CRSM reader 345 such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the  
5 signal bus 310 for reading a CRSM 350 such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the system 100 may receive programs and data via the CRSM reader 345.

FIG. 4 is a block diagram illustrating details of the remote client 120. The client 120 includes a processor 405, such as an Intel Pentium® microprocessor or a  
10 Motorola Power PC® microprocessor, coupled to a communications channel 410. The client 120 further includes an input device 415 such as a keyboard and mouse, an output device 420 such as a CRT display, data storage 425 such as a magnetic disk, and internal storage 430 such as RAM, each coupled to the communications channel 410. A communications interface 425 couples the communications  
15 channel 410 to the computer network 125.

An operating system 440 controls processing by processor 405, and is typically stored in data storage 430 and loaded into internal storage 435 (as illustrated) for execution. The PIM 170 and assistant 185 may be stored in data storage 430, and loaded into internal storage 435 (as illustrated) for execution by  
20 the processor 405.

One skilled in the art will recognize that the system 100 may also include additional information, such as network connections, additional memory, additional processors, LANs, input/output lines for transferring information across a hardware channel, the Internet or an intranet, etc. One skilled in the art will also  
5 recognize that the programs and data may be received by and stored in the system 100 in alternative ways. For example, a CRSM reader 445 such as a magnetic disk drive, hard disk drive, magneto-optical reader, CPU, etc. may be coupled to the signal bus 310 for reading a CRSM 450 such as a magnetic disk, a hard disk, a magneto-optical disk, RAM, etc. Accordingly, the system 100 may receive  
10 programs and data via the CRSM reader 445.

FIG. 5 is a block diagram illustrating a PIM interface 500, which includes a header 505 and a selection window 510.

The header 505 includes a synchronize button 540 and a "borrow me" button 545, which are presented by the assistant 175, 180 or 185 incorporated in  
15 the PIM 160, 165 or 170. Invoking the synchronize button 540 causes the assistant 175, 180 or 185 to enable synchronization of data entered into the PIM 160, 165 or 170 with the workspace data 135 on the global server 135. The synchronize button 540 may enable the user to configure a preference file that indicates when automatic synchronization is to initiate and may also enable a user to effect manual  
20 synchronization.

5 The "borrow me" button 545 enables a user to use a PIM 160, 165 or 170 for viewing and manipulating workspace data 135 downloaded from the global server 105. That is, invoking the "borrow me" button 545 causes the corresponding assistant 175, 180 or 185 to communicate with the global server 105, to provide user identification and authentication information to the global server 105, to download workspace data 135 from the global server 105, to display and enable manipulation of the downloaded data 135 using the PIM interface 500, and to synchronize the manipulated downloaded data 135 upon logout. Since the PIM interface 500 is provided by the pre-existing PIM, the assistant 175, 180 or 185 need not provide its own data interface. Only a single interface is needed.

10 It will be appreciated that upon logout, the base systems 140 and 150 will cooperate with the synchronization agent 130 to synchronize automatically the workspace data 135 on the global server 105 with the workspace data 145 and 155. Accordingly, the user always has access to the most updated versions of workspace data from any site that executes a PIM 160, 165 or 170 having an assistant 175, 180 or 185 embodied therein.

15 It will be appreciated that the synchronize button 540 is most helpful to the work client 110 and the home client 115, since typically the work client 110 and home client 115 will set the preference file to configure automatic synchronization. Synchronization of the manipulated workspace data 135 at the remote client 120 will most often be effected through the automatic logout procedures of the "borrow



me” button. Logout is described in greater detail with reference to the Outlook™ and Lotus Organizer examples shown and described below with reference to FIG.

7. Accordingly, the borrow me button 545 is most helpful to the remote client 120.

The selection window 510 provides a list of buttons 507, wherein each  
5 button 507 corresponds to a set of workspace elements, e.g., e-mails 515, contacts 520, files 525, calendar data 530 and bookmarks 535. A mouse-down on a virtual button 507 causes the selection of a corresponding workspace element set and the selection of a corresponding user interface for displaying and enabling manipulation of the workspace elements included in the set. For example,  
10 selection of button 515 selects the e-mail set, and selects a corresponding user interface for displaying, writing, forwarding, etc. e-mails. Selecting a button 507 causes the assistant 175, 180 or 185 to download the corresponding workspace data 135, and causes the PIM 160, 165 or 170 to display and enable manipulation of the downloaded data 135 on a workspace element set interface (shown and  
15 described with reference to FIG. 6).

FIG. 6 illustrates an example e-mail workspace element set user interface 600 (commonly referred to as the “In-Box”) for displaying received e-mails. The user interface 600 includes a header 605, an e-mail list window 610 and a manipulation command window 650.

20 The header 605 lists the name of the workspace element set, namely, “E-Mail.” The e-mail list window 610 comprises three columns, including an origin

column 615 which provides the origin of each e-mail, a subject column 620 which provides the subject of each e-mail, and a date column 625 which provides the date each e-mail was received. The e-mail list window 610 may display e-mails stored in a local e-mail database (not shown), e-mails stored in the e-mail server (not shown) or e-mails downloaded from the global server 105. The e-mails shown include a first e-mail from Joe Smith, a second e-mail from Tom Jones, and a third e-mail from Roy White. If the user depressed the "borrow me" button 545 shown in FIG. 5, then the e-mail list displayed would be the e-mails stored and downloaded from the global server 105.

The manipulation window 650 includes available functions such as the conventional e-mail read function 630, e-mail reply function 635, e-mail forward function 640 and new e-mail write function 645. It will be appreciated that the columns and functions will vary based on the PIM.

FIG. 7 is a block diagram illustrating details of a generic assistant 700, which exemplifies each of the assistant 175, 180 and 185. The generic assistant 700 includes a communications module 705, locator modules 710, a general synchronization module 715, a content-based synchronization module 720, a security module 725, an instantiator 730, a data reader 735, a PIM Application Program Interface (API) 740 and a de-instantiator 745. The synchronization function of the assistant 700 uses the communications module 705, the locator modules 710, the general synchronization module 715, the content-based

synchronization module 720, the security module 725 and the PIM API 740. The "borrow me" function of the assistant 700 uses the communications module 705, the locator modules 710, the security module 725, the instantiator 730, the data reader 735, the PIM API 740 and the de-instantiator 745.

5           The communications module 705 includes routines for compressing and decompressing data, and routines for communicating with the synchronization agent 130. The communications module 705 may apply Secure Socket Layer (SSL) technology to establish a secure communication channel. Examples of communications modules 705 may include TCP/IP stacks or the AppleTalk  
10   protocol.

          The locator modules 710 include routines for identifying the memory locations of the workspace elements in the workspace data 135. Workspace element memory location identification may be implemented using intelligent software, i.e., preset memory addresses or the system's registry, or using dialogue  
15   boxes to query the user. Accordingly, the locator modules 710 determine the memory addresses of the workspace elements in e-mail workspace data 135, in file workspace data 135, in calendar workspace data 135, etc.

          The general synchronization module 715 examines the workspace data 135 on the global server 105 to determine whether it had been modified while the user  
20   manipulated the data on the client 110, 115 or 120. Further, the general synchronization module 715 determines whether the user manipulated any data on

the client 110, 115 or 120. If the general synchronization module 715 determines that only the data on the client 110, 115 or 120 was manipulated, then the general synchronization module 715 computes and sends the changes to the synchronization agent 130 of the global server 105. The general synchronization module 715 is initiated when the synchronization button 540 is depressed and during the logout procedures of the "borrow me" function.

The synchronization agent 130 then updates a last synchronization signature to indicate to all base systems 140 and 150 that synchronization with workspace data 145 and synchronization with workspace data 155 are needed. If the general synchronization module 715 determines that changes were made only to the workspace data 135 on the global server 105, then the general synchronization module 715 instructs the synchronization agent 130 to compute and transmit the changes made to the client 110, 115 or 120 at the client's request. The client 110 or 120 then updates its information. It will be appreciated that sending only the changes reduces processor load and increases transmission line efficiency, although alternatively an entire manipulated workspace element can be sent to the global server 105.

If the general synchronization module 715 determines that the workspace data 135 on the global server 105 has been modified since download, and that the data on the client 110, 115 or 120 has been modified, then the general synchronization module 715 instructs the content-based synchronization module

720 to perform its duties. The content-based synchronization module 720 includes routines for reconciling two or more modified versions of a workspace element.

The content-based synchronization module 720 may request a user to select the preferred one of the modified versions or may respond based on preset

5 preferences, i.e., by storing both versions in both stores or by integrating the changes into a single preferred version which replaces each modified version at both stores.

The security module 725 includes routines for obtaining user identification and authentication using such techniques as obtaining login and password  
10 information, obtaining a response to a challenge, obtaining a public key certificate, etc. The security module 725 performs identification and authentication techniques to confirm authorization by the user to access the workspace data 135 stored on the global server 105. It will be appreciated that authorization may be granted only to portion of the workspace data 135 that belongs to the user.

15 The instantiator 730 is an application program interface 730 that creates a window for displaying and enabling manipulation of the workspace data 135 downloaded from the global server 105. In an object-oriented environment, the instantiator 730 may create a new instance for the workspace data 135.

Alternatively, the instantiator 730 may store the local data to a buffer (not shown)  
20 and use the current interface to display and enable manipulation of the workspace data 135.

The data reader 735 communicates with the synchronization agent 130 at the global server 105, and retrieves the workspace data 135 requested. For example, if the user depresses the "borrow me" button 545 (FIG. 5) and depresses the e-mail button 515, then the data reader 735 retrieves the e-mail workspace elements of the workspace data 135, and delivers them to the PIM API 740.

The PIM API 740 translates and transfers the workspace data 135 received from the global server 105 to the PIM 160, 165 or 170 for display and enabling manipulation thereto. The PIM API 740 further translates and transfers the workspace data manipulated on the client 110, 115 or 120 from the PIM 160, 165 or 170 back to the global server 105.

The de-instantiator 745 returns the PIM 160, 165 or 170 to the state before the user selected the "borrow me" button 545. The user may initiate operations of the de-instantiator 745 by depressing an "unborrow me" button (not shown) that is presented after selection of the "borrow me" button 545. The de-instantiator 745 deletes any instance created by the instantiator 730, deletes all workspace data 135 and data created by the user on the client 110, 115 or 120 and automatically initiated synchronization of any manipulated downloaded data 135 with the workspace data 135 stored at the global server 105.

Operations of the instantiator 730, the data reader 735, the PIM API 740 and the de-instantiator 745 are described in greater detail with reference to the following examples:

OUTLOOK EXAMPLE:

| <u>Action</u>        | <u>Global Data</u>  | <u>Local Data</u>                 |
|----------------------|---|-----------------------------------|
| standby              | —   | local data → pst <sup>local</sup> |
| button depressed     | —   | pst <sup>local</sup>              |
| enter login/password | —   | pst <sup>local</sup>              |
| authenticate         | —   | pst <sup>local</sup>              |
| send global data     | global data → pst <sup>local</sup>  | local data → pst buffer           |
| manipulate data      | global data → global data 2   | pst <sup>buffer</sup>             |
| logout               | 1) Compute $\Delta$ global data<br><br>2) Synchronize $\Delta$ global data<br>with global server<br><br>3) Delete global data 2<br><br>4) | local data → pst <sup>local</sup> |

As illustrated by the Outlook<sup>TM</sup> example above, during standby, the PIM 160, 165 or 170 stores the local data on the client 110, 115 or 120 in a personal folder store pst<sup>local</sup>. The user then depresses the “borrow me” button 545. The security module 725 requests the user to enter a login and password, which the

global server 105 authenticates. During these steps, it will be appreciated that the local data remains stored in  $pst^{local}$ . Upon user identification and authentication, the global server 105 sends the workspace data 135 (global data) to the requesting client 110, 115 or 120. The instantiator 730 on the client 110, 115 or 120 transfers  
5 the local data from  $pst^{local}$  to a buffer  $pst^{buffer}$ , and stores the received global data into  $pst^{local}$ . The data reader 745 and PIM API 740 enable the user to manipulate the global data, the manipulated data being referred to herein as "global data 2." Upon logout, for example, after an "unborrow me" button (not shown) is depressed, the global data 2 is synchronized with the workspace data  
10 135. Namely, the general synchronization module 715 determines the changes made ( $\Delta$ global data), and synchronizes  $\Delta$ global data with the workspace data 135. The de-instantiator 745 deletes global data 2 and  $\Delta$ global data, and returns the local data to  $pst^{local}$ .

09008354-011698



LOTUS ORGANIZER EXAMPLE:

| <u>Action</u>        | <u>Global Data</u>  | <u>Local Data</u> |
|----------------------|---|-------------------|
| standby              | —   | local.org         |
| button               | —   | local.org         |
| enter login/password | —   | local.org         |
| authenticate         | new instance  | local.org         |
| send global data     | open with global.org  | local.org         |
| manipulate data      | global.org → global.org <sup>2</sup>  | local.org         |
| logout               | 1) compute $\Delta$ global.org<br>2) Synchronize $\Delta$ global.org<br>with global server<br>3) delete global.org <sup>2</sup> | local.org         |

As illustrated by the Lotus Organizer example above, during standby, the

PIM 160, 165 or 170 stores the local data on the client 110, 115 or 120 in

- 5 local.org. The user then depresses the “borrow me” button 545. The security module 725 requests the user to enter a login and password, which the global server 105 authenticates. During these steps, it will be appreciated that the local data remains stored in local.org. Upon user identification and authentication, the

global server 105 sends the workspace data 135 (global data) to the requesting client 110, 115 or 120. The instantiator 730 on the client 110, 115 or 120 creates a new instance, e.g., a new window, of PIM API 740 and stores the received global data into another file, i.e., global.org. The data reader 745 and PIM API 740

5 enable the user to manipulate the global data, the manipulated data being referred to herein as "global data 2." Upon logout, the global data 2 is synchronized with the workspace data 135. Namely, the general synchronization module 715 determines the changes made ( $\Delta$ global data), and synchronizes  $\Delta$ global data with the workspace data 135. The de-instantiator 745 deletes global data 2,  $\Delta$ global

10 data and global.org.

FIG. 8 is a flowchart illustrating a method 800 of accessing data remotely in accordance with the present invention. The method 800 begins with the processor 405 in step 805 opening the PIM 160, 165 or 170 per user request, and the PIM 160, 165 or 170 opening a PIM interface 500 (FIG. 5). The PIM 160, 165 or 170

15 in step 810 receives a "borrow me" request from the user, i.e., the user depresses the "borrow me" button 545. The PIM API 740 in step 815 recognizes the request, and instructs the communications module 705 to create a communications link with the global server 105.

The security module 725 in step 820 requests and transmits identification and authentication information such as login and password information from the

20 user to the global server 105 for examination. If the global server 105 fails to

identify or authenticate the user, then the method 800 ends. Otherwise, the instantiator 730 in step 825 opens a PIM interface 500 to display and enable manipulation of the workspace data 135 downloaded from the global server 105. The data reader 735 in step 830 reads the workspace data 135 downloaded from the global server 105, and in step 835 translates the data to the appropriate format if necessary. That is, the data reader 735 translates the workspace data 135 from the format implemented by the global server 105 to the format implemented by the PIM 160, 165 or 170. The PIM API 740 in step 840 passes the translated workspace data 135 to the PIM interfaces 500 and 600.

The PIM 160, 165 or 170 enables the user in step 845 to manipulate the workspace data 135 as necessary. Manipulation includes adding new data, deleting workspace data 135, editing workspace data 135, etc. For example, the user can depress the e-mail button 515 in interface 500 to select, review and manipulate e-mail in interface 600, and then can depress the calendar button 530 in interface 500 to select, review and manipulate calendar information (not shown) in an interface similar to the e-mail interface 600. In step 850, the PIM API 740 waits to receive an "end session" request. Until an "end session" request is received, the method 800 returns to step 830 to enable continued data review and manipulation.

Upon receiving an "end session" or "unborrow me" request, the de-instantiator 745 initiates the general synchronization module 715 in step 855 to

27

synchronize the manipulated workspace data on the client 110, 115 or 120 with the workspace data 135 on the global server 105, if required. Synchronization is described in greater detail with reference to FIG. 9. The de-instantiator 745 in step 860 deletes the workspace data on the client 110, 115 or 120, and deletes all  
5 records of the matter. Method 800 then ends.

FIG. 9 is a flowchart illustrating a method 900 for synchronizing workspace data in a computer network 100. Method 900 begins with the communications module 705 in step 905 establishing a communications link with the synchronization agent 130 of the global server 105. The locator modules 710 in  
10 step 910 identify the memory locations of the workspace elements in the workspace data 135. It will be appreciated that workspace element memory location identification may be implemented using intelligent software or dialogue boxes.

The general synchronization module 715 in step 915 compares version  
15 information (not shown) for each workspace element in the workspace data (on the client 110, 115 or 120 and on the global server 105) against a last synchronization signature to determine which workspace elements have been modified. In this embodiment, a workspace element may have been modified if the date and time of the last modification is after the date and time of the downloading.

20 If the general synchronization module 715 locates no modified workspace elements in the workspace data on the client 110, 115 or 120, then the method 900

28

ends. Otherwise, the general synchronization module in step 920 determines whether the version of the same workspace element of the workspace data 135 on the global server 105 has been modified since the data 135 was downloaded.

If only the version on the client 110, 115 or 120 has been modified, then the  
5 general synchronization module 715 in step 925 stores the updated version of the workspace element at the global server 105. To store the updated version on the global server 105, the general synchronization module 715 may compute the changes made and forward the changes to the synchronization agent 130. The synchronization agent 130 enters the changes into the global server 105 version.  
10 The general synchronization module 715 in step 930 determines whether all workspace elements downloaded to the client 110, 115 or 120 have been examined. If not, then method 900 returns to step 915. Otherwise, the synchronization agent 130 in step 935 updates the last synchronization signature, and method 900 ends. Updating the last synchronization signature will instruct the  
15 base systems 140 and 150 to synchronize the workspace data 145 and 155 with the workspace data 135 on the global server 105, as described in the patent applications incorporated by reference above.

If the general synchronization module 715 in step 920 determines that both the version on the client 110, 115 or 120 and the version on the global server 105  
20 have been modified, then the general synchronization module in step 935 instructs the content-based synchronization module 729 to reconcile the modified versions.

29

Reconciliation may include requesting instructions from the user, or performing based on pre-selected preferences responsive actions such as storing both versions at the global server 105. The general synchronization module 715 in step 940 stores the preferred version on the global server 105. Method 900 then proceeds to  
5 step 930.

The foregoing description of the preferred embodiments of the present invention is by way of example only, and other variations and modifications of the above-described embodiments and methods are possible in light of the foregoing teaching. Although the network sites are being described as separate and distinct  
10 sites, one skilled in the art will recognize that these sites may be a part of an integral site, may each include portions of multiple sites, or may include combinations of single and multiple sites. Further, components of this invention may be implemented using a programmed general purpose digital computer, using application specific integrated circuits, or using a network of interconnected  
15 conventional components and circuits. Connections may be wired, wireless, modem, etc. The embodiments described herein are not intended to be exhaustive or limiting. The present invention is limited only by the following claims.